

On efficient constructions of short lists containing mostly Ramsey graphs

Marius Zimand *

Abstract

One of the earliest and best-known application of the probabilistic method is the proof of existence of a $2 \log n$ -Ramsey graph, i.e., a graph with n nodes that contains no clique or independent set of size $2 \log n$. The explicit construction of such a graph is a major open problem. We show that a reasonable hardness assumption implies that in polynomial time one can construct a list containing $\text{polylog}(n)$ graphs such that most of them are $2 \log n$ -Ramsey.

1 Introduction

A k -Ramsey graph is a graph G that has no clique of size k and no independent set of size k . It is known that for all sufficiently large n , there exists a $2 \log n$ -Ramsey graph with n vertices. The proof is nonconstructive, but of course such a graph can be built in exponential time by exhaustive search. A major line of research is dedicated to constructing a k -Ramsey graph with n vertices with k as small as possible and in time that is bounded by a small function in n , for example in polynomial time, or in quasi-polynomial time, $\text{DTIME}[2^{\text{polylog}(n)}]$. Till recently, the best polynomial-time construction of a k -Ramsey graph with n vertices has been the one by Frankl and Wilson [FW81], for $k = 2^{\tilde{O}(\sqrt{\log n})}$. Using deep results from additive combinatorics and the theory of randomness extractors and dispersers, Barak, Rao, Shaltiel and Wigderson [BRSW06] improved this to $k = 2^{(\log n)^{o(1)}}$. Notice that this is still far off from $k = 2 \log n$.

As usual when dealing with very difficult problems, it is natural to consider easier versions. In this case, one would like to see if it is possible to efficiently construct a small list of n -vertices graphs with the guarantee that one of them is $2 \log n$ -Ramsey. The following positive results hold.

*Department of Computer and Information Sciences, Towson University, Baltimore, MD.; email: mzimand@towson.edu; <http://triton.towson.edu/~mzimand>. This work has been supported by NSF grant CCF 1016158.

Theorem 1.1 *There exists a quasipolynomial-time algorithm that on input 1^n returns a list with $2^{\log^3 n}$ graphs with n vertices, and most of them are $2 \log n$ -Ramsey. In fact, since in quasipolynomial time one can check whether a graph is $2 \log n$ -Ramsey, the algorithm can be modified to return one graph that is $2 \log n$ -Ramsey.*

Theorem 1.2 *Under a reasonable hardness assumption H , there exists a constant c and a polynomial-time algorithm that on input 1^n returns a list with $\log^c n$ graphs with n vertices, and most of them are $2 \log n$ -Ramsey.*

The proofs of these two results use basic off-the-shelf derandomization techniques. The proof (one of them) of Theorem 1.1 notices that the probabilistic argument that shows the existence of $2 \log n$ -Ramsey graphs only needs a distribution on the set of n -vertices graphs that is $2 \log^2 n$ -wise independent. There exist such distributions whose support have the following properties: (a) the size is $2^{O(\log^3 n)}$ and (b) it can be indexed by strings of size $O(\log^3 n)$. Therefore if we make an exhaustive search among these indeces, we obtain the result.

Theorem 1.2 uses a pseudo-random generator g that can fool NP-predicates. The assumption H , which states that there exists a function in E that, for some $\epsilon > 0$, requires circuits with SAT gates of size $2^{\epsilon n}$, implies the existence of such pseudo-random generators. Then going back to the previous proof, it can be observed that the property that an index corresponds to a graph that is not $2 \log n$ -Ramsey is an NP predicate. Since most indeces correspond to graphs that are $2 \log n$ -Ramsey, it follows that for most seeds s , $g(s)$ is also $2 \log n$ -Ramsey. Therefore, it suffices to make an exhaustive search among all possible seeds. Since a seed has length $O(\log |\text{index}|) = O(\log \log^3 n)$, the result follows.

Theorem 1.2 can be strengthened to produce a list of concise representations of graphs. A string t is a *concise representation* of a graph $G = (V, E)$ with $V = \{1, \dots, n\}$ if there is an algorithm A running in time $\text{poly}(\log n)$ such that for every $u \in V, v \in V$, $A(t, u, v) = 1$ if $(u, v) \in E$ and $A(t, u, v) = 0$ if $(u, v) \notin E$. With basically the same proof as that of Theorem 1.2 one can show the following result.

Theorem 1.3 *Under a reasonable hardness assumption H , there exists a constant c and an algorithm running in time $\text{poly}(\log n)$ that on input n (written in binary notation) returns a list $t_1, \dots, t_{\log^c n}$, and most elements of the list are concise representations of $2 \log n$ -Ramsey graphs.*

Theorem 1.1 is folklore. It appears implicitly in the paper of M. Naor [Nao92]. Theorem 1.2 may also be known, but we are not aware of any published statement of it. Fortnow in the *Computational Complexity* blog [For06] and Santhanam at the 2011 Bertinoro seminar on Ramsey theory [San11] mention a weaker version of Theorem 1.2, in which the same hardness assumption is used but the size of the list is polynomial instead of polylogarithmic. This motivated us to write this note.

Section 4 contains some additional remarks. First we analyze the implication of Theorem 1.2 when plugged in a construction of M. Naor [Nao92] that builds a

k -Ramsey graph from a list of graphs, most of which are k' -Ramsey graphs, which is exactly what Theorem 1.2 delivers. We notice that the parameters obtained in this way are inferior to the result of Barak et al. [BRSW06]. Secondly, we consider the problem of explicit lower bounds for the van der Waerden Theorem, a problem which is related to the explicit construction of Ramsey graphs. We notice that the hardness assumption which derandomizes BPP implies lower bounds for the van der Waerden Theorem that match the non-constructive lower bounds obtained via the Lovasz Local Lemma. The original proof of the Lovasz Local Lemma does not seem to yield this result. Instead we use a proof of Gasarch and Haeupler [GH11], based on the methods of Moser [Mos09] and Moser and Tardos [MT10].

2 The hardness assumption

The hardness assumption needed in theorem 1.2 is that there exists a function f computable in E (where $E = \bigcup_c \text{DTIME}[2^{cn}]$) that, for some $\epsilon > 0$, cannot be computed by circuits of size $2^{\epsilon n}$ that also have SAT gates (in addition to the standard logical gates). More formally let us denote by $C_f^{\text{SAT}}(n)$ the size of the smallest circuit with SAT gates that computes the function f for inputs of length n .

Assumption H : There exists a function f in E such that, for some $\epsilon > 0$, for every n , $C_f^{\text{SAT}}(n) > 2^{\epsilon n}$.

Klivans and van Melkebeek [KvM02], generalizing the work of Nisan and Wigderson [NW94] and Impagliazzo and Wigderson [IW97], have shown that, under assumption H , for every k , there is a constant c and a pseudo-random generator $g : \{0, 1\}^{c \log n} \rightarrow \{0, 1\}^n$, computable in time polynomial in n , that fools all n^k -size circuits with SAT gates. Formally, for every circuit C with SAT gates of size n^k ,

$$|\text{Prob}_{s \in \{0,1\}^{c \log n}}[C(g(s)) = 1] - \text{Prob}_{z \in \{0,1\}^n}[C(z) = 1]| < 1/n^k.$$

We note that assumption H is realistic. Miltersen [Mil01] has shown that it is implied by the following natural assumption, involving uniform complexity classes: for every $\epsilon > 0$, there is a function $f \in E$ that cannot be computed in space $2^{\epsilon n}$ for infinitely many lengths n .

3 Proofs

Proof of Theorem 1.1.

Let us first review the probabilistic argument showing the existence of $2 \log n$ -Ramsey graphs. A graph G with n vertices can be represented by a string of length $\binom{n}{2}$. If we take at random such a graph and fix a subset of k vertices, the probability

that the set forms a clique or an independent set is $2^{-\binom{k}{2}+1}$. The probability that this holds for some k -subset is bounded by

$$\begin{aligned} \binom{n}{k} \cdot 2^{-\binom{k}{2}+1} &\leq \left(\frac{en}{k}\right)^k \cdot 2^{-\binom{k}{2}+1} \\ &= 2^{k \log \frac{en}{k} - \binom{k}{2} + 1}. \end{aligned}$$

For $k = 2 \log n$, the above expression goes to 0. Thus, for n large enough, the probability that a graph G is $2 \log n$ -Ramsey is ≥ 0.99 .

The key observation is that this argument remains valid if we take a distribution that is $2 \log^2 n$ -wise independent. Thus, we can take a polynomial $p(X)$ of degree $2 \log^2 n$ over the field $\text{GF}[2^q]$, where $q = \log \binom{n}{2}$. To the polynomial p we associate the string $\tilde{p} = p(a_1)_1 \dots p(a_{\binom{n}{2}})_1$, where $a_1, \dots, a_{\binom{n}{2}}$ are the elements of the field and $(p(a))_1$ is the first bit of $p(a)$. When p is random, this yields a distribution over strings of length $\binom{n}{2}$ that is $2 \log^2 n$ -wise independent. Observe that a polynomial p is given by a string of length $\bar{n} = (2 \log^2 n + 1) \log \binom{n}{2} = O(\log^3 n)$. It follows that

$$\text{Prob}_{p \in \{0,1\}^{\bar{n}}}[\tilde{p} \text{ is } 2 \log n\text{-Ramsey}] \geq 0.99.$$

In quasipolynomial time we can enumerate the graphs \tilde{p} , and 99% of them are $2 \log n$ -Ramsey. ■

Note. By using an almost k -wise independent distribution (see [NN93, AGHR92]), one can reduce the size of the list to $2^{O(\log^2 n)}$.

Proof of Theorem 1.2 and of Theorem 1.3.

Let p, \tilde{p}, \bar{n} be as in the proof of Theorem 1.1. Thus:

- $p \in \{0,1\}^{\bar{n}}$ represents a polynomial,
- \tilde{p} is built from the values taken by p at all the elements of the underlying field, and represents a graph with n vertices,
- $\bar{n} = O(\log^3 n)$.

Let us call a string p *good* if \tilde{p} is a $2 \log n$ -Ramsey graph.

Checking that a string p is not *good* is an NP predicate. Indeed, p is not *good* iff $\exists (i_1, \dots, i_{2 \log n}) \in [n]^{2 \log n}$ [vertices $i_1, \dots, i_{2 \log n}$ in \tilde{p} form a clique or an independent set]. The \exists is over a string of length polynomial in $|p|$ and the property in the right parentheses can be checked by computing $O(\log^2 n)$ values of the polynomial p , which can be done in time polynomial in $|p|$.

Assumption H implies that there exists a pseudo-random generator $g : \{0,1\}^{c \log \bar{n}} \rightarrow \{0,1\}^{\bar{n}}$, computable in time polynomial in \bar{n} , that fools all NP predicates, and, in particular, also the one above. Since 99% of the p are *good*, it follows that for 90% of the seeds $s \in \{0,1\}^{c \log \bar{n}}$, $g(s)$ is *good*, i.e., for 90% of s , $\widehat{g(s)}$ is $2 \log n$ -Ramsey. Note that from a seed s we can compute $g(s)$ and next $\widehat{g(s)}$ in

time polynomial in n . If we do this for every seed $s \in \{0, 1\}^{c \log \bar{n}}$, we obtain a list with $\bar{n}^c = O(\log^{3c} n)$ graphs of which at least 90% are $2 \log n$ -Ramsey graphs.

Theorem 1.3 is obtained by observing that $\{g(s) \mid s \in \{0, 1\}^{c \log \bar{n}}\}$ is a list that can be computed in $\text{poly}(\log n)$ time, and most of its elements are concise representations of $2 \log n$ -Ramsey graphs. ■

4 Additional remarks

4.1 Constructing a single Ramsey graph from a list of graphs of which the majority are Ramsey graphs

M. Naor [Nao92] has shown how to construct a Ramsey graph from a list of m graphs such that all the graphs in the list, except at most αm of them, are k -Ramsey. We analyze what parameters are obtained, if we apply Naor's construction to the list of graphs in Theorem 1.2.

The main idea of Naor's construction is to use the product of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, which is the graph whose set of vertices is $V_1 \times V_2$ and edges defined as follows: there is an edge between (u_1, u_2) and (v_1, v_2) if and only if $(u_1, v_1) \in E_1$ or $(u_1 = v_1)$ and $(u_2, v_2) \in E_2$. Then, one can observe that if G_1 is k_1 -Ramsey and G_2 is k_2 -Ramsey, the product graph, $G_1 \times G_2$ is $k_1 k_2$ -Ramsey. Extending to the product of multiple graphs G_1, G_2, \dots, G_m where each G_i is k_i -Ramsey, we obtain that the product graph is $k_1 k_2 \dots k_m$ -Ramsey.

If we apply this construction to a list of m graphs G_1, G_2, \dots, G_m , each having n vertices and such that $\text{Prob}_i[G_i \text{ is not } k\text{-Ramsey}] \leq \alpha$, we obtain that the product of G_1, G_2, \dots, G_m is a graph G with $N = n^m$ vertices that is t -Ramsey for $t = n^{\alpha m} k^{(1-\alpha)m}$. For $\alpha \leq 1/\log n$, we have $t \leq (2k)^m$. The list produced in Theorem 1.2 has $m = \log^c n$, $k = 2 \log n$, and one can show that $\alpha \leq 1/\log n$. The product graph G has $N = 2^{\log^c n \log n}$ vertices and is t -Ramsey for $t \leq 2^{\log^c n \cdot \log \log n + O(1)} < 2^{(\log N)^{1-\beta}}$, for some positive constant β .

Thus, under assumption H , there is a positive constant β and a polynomial time algorithm that on input 1^N constructs a graph with N vertices that is $2^{(\log N)^{1-\beta}}$ -Ramsey. Note that this is inferior to the parameters achieved by the unconditional construction of Barak, Rao, Shaltiel and Wigderson [BRSW06].

4.2 Constructive lower bounds for the van der Waerden theorem

Van der Waerden Theorem is another classical result in Ramsey theory. It states that for every c and k there exists a number n such that for any coloring of $\{1, \dots, n\}$ with c colors, there exists k elements in arithmetic progression (k -AP) that have the

same color. Let $W(c, k)$ be the smallest such n . One question is to find a constructive lower bound for $W(c, k)$. To simplify the discussion, let us focus on $W(2, k)$.

In other words, the problem that we want to solve is the following:

For any k , we want to find a value of $n = n(k)$ as large as possible and a 2-coloring of $\{1, \dots, n\}$ such that no k -AP is monochromatic. Furthermore, we want the 2-coloring to be computable in time polynomial in n .

Gasarch and Haeupler [GH11] have studied this problem. They present a probabilistic polynomial time construction for $n = \frac{2^{k-1}}{ek} - 1$ (i.e., the 2-coloring is obtained by a probabilistic algorithm running in $2^{O(k)}$ time) and a (deterministic) polynomial time construction for $n = \frac{2^{(k-1)(1-\epsilon)}}{4k}$ (i.e., the 2-coloring is obtained in deterministic $2^{O(k/\epsilon)}$ time). Their constructions are based on the constructive version of the Lovasz Local Lemma due to Moser [Mos09] and Moser and Tardos [MT10]. The probabilistic algorithm of Gasarch and Haeupler is “BPP-like”, in the sense that it succeeds with probability $2/3$ and the correctness of the 2-coloring produced by it can be checked in polynomial time. It follows that it can be derandomized under the hardness assumption that derandomizes BPP, using the Impagliazzo-Wigderson pseudo-random generator [IW97]. It is interesting to remark that the new proof by Moser and Tardos of the Local Lovasz Lemma is essential here, because the success probability guaranteed by the classical proof is too small to be used in combination with the Impagliazzo-Wigderson pseudo-random generator.

We proceed with the details.

We use the following hardness assumption H' (weaker than assumption H), which is the one used to derandomize BPP [IW97].

Assumption H' : There exists a function f in E such that, for some $\epsilon > 0$, for every n , $C_f(n) > 2^{\epsilon n}$.

Impagliazzo and Wigderson [IW97] have shown that, under assumption H , for every k , there is a constant c and a pseudo-random generator $g : \{0, 1\}^{c \log n} \rightarrow \{0, 1\}^n$ that fools all n^k -size circuits and that is computable in time polynomial in n .

Proposition 4.1 *Assume assumption H' . For every k , let $n = n(k) = \frac{2^{k-1}}{ek} - 1$. There exists a polynomial-time algorithm that on input 1^n 2-colors the set $\{1, \dots, n\}$ such that no k -AP is monochromatic.*

Proof. The algorithm of Gasarch and Haeupler [GH11], on input 1^n , uses a random string z of size $|z| = n^c$, for some constant c , and, with probability at least $2/3$, succeeds to 2-color the set $\{1, \dots, n\}$ such that no k -AP is monochromatic. Let us call a string z to be *good* for n if the Gasarch-Haeupler algorithm on input 1^n and randomness z , produces a 2-coloring with no monochromatic k -APs. Note that there exists a polynomial-time algorithm A that checks if a string z is *good* or not, because the Gasarch-Haeupler algorithm runs in polynomial time and the number of k -APs inside $\{1, \dots, n\}$ is bounded by n^2/k . Using assumption H' and invoking the result

of Impagliazzo and Wigderson [IW97], we derive that there exists a constant d and a pseudo-random generator $g : \{0, 1\}^{d \log n} \rightarrow \{0, 1\}^{n^c}$ such that

$$\text{Prob}_{s \in \{0, 1\}^{d \log n}}[A(g(s)) = \text{good for } n] \geq 2/3 - 1/10 > 0.$$

Therefore if we try all possible seeds s of length $d \log n$, we will find one s such that $g(s)$ induces the Gasarch-Haeupler algorithm to 2-color the set $\{1, \dots, n\}$ such that no k -AP is monochromatic. ■

References

- [AGHR92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k -wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.
- [BRSW06] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl-Wilson construction. In Jon M. Kleinberg, editor, *STOC*, pages 671–680. ACM, 2006.
- [For06] L. Fortnow. Full derandomization. Computational Complexity blog, July 31 2006, 2006.
- [FW81] P. Frankl and R. M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.
- [GH11] William I. Gasarch and Bernhard Haeupler. Lower bounds on van der Waerden numbers: Randomized- and deterministic-constructive. *Electr. J. Comb.*, 18(1), 2011.
- [IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 220–229, New York, May 1997. Association for Computing Machinery.
- [KvM02] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.
- [Mil01] P. B. Miltersen. Derandomizing complexity classes. In P. Pardalos, J. Reif, and J. Rolim, editors, *Handbook on Randomized Computing, Volume II*. Kluwer Academic Publishers, 2001.
- [Mos09] Robin A. Moser. A constructive proof of the Lovász local lemma. In Michael Mitzenmacher, editor, *STOC*, pages 343–350. ACM, 2009.
- [MT10] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2), 2010.

- [Nao92] M. Naor. Constructing Ramsey graphs from small probability spaces. Technical report, IBM Research Report RJ 8810 (70940), 1992.
- [NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, August 1993.
- [NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.
- [San11] R. Santhanam. Communication at RaTLocc, Bertinoro, May 22-27, 2011.